COSE474-2024F: Final Project
"Enhancing Semiconductor Defect Detection through Single-modal Deep Learning Frameworks"

Jung Ha Yeon

1. Introduction

1.1. Motivation

Semiconductors are the backbone of modern technology, playing a pivotal role in various industries, including artificial intelligence, consumer electronics, and automotive systems. Recently, my personal interest in semiconductors has significantly grown, especially after exploring cutting-edge AI semiconductor technologies such as NVIDIA's advancements in GPU-based AI systems. These developments sparked a curiosity to contribute to the semiconductor field by leveraging artificial intelligence techniques. This natural progression led me to focus on creating a model that could enhance quality control processes in semiconductor manufacturing. By analyzing the WM-811K wafer map dataset, I aim to develop a robust framework that identifies and categorizes defect patterns, ensuring higher efficiency and reliability in the production pipeline.

1.2. Problem Definition

The WM-811K wafer map dataset provides visual representations of wafers with defect patterns and corresponding defect type labels, commonly referred to as *failures*. Analyzing these failure types is critical for understanding the underlying causes of defects and improving manufacturing processes. The primary goal of this project is to utilize the WM-811K dataset to not only detect anomalies but also categorize them based on specific failure types. This involves building a machine learning pipeline capable of accurately classifying wafers into their respective defect categories, which include both typical defect types and rarer anomaly patterns.

1.3. Concise Description of Contribution

This project leverages the pre-trained Contrastive Language-Image Pretraining (CLIP) model's visual encoder to address the challenge of wafer defect classification and anomaly detection. Key contributions include:

- Development of a custom preprocessing pipeline for the WM-811K dataset to extract and standardize defect pattern information.
- Application of the CLIP visual encoder to learn representations of wafer defect patterns and perform multi-class classification of failure types.
- Integration of supervised learning techniques to enhance the accuracy of anomaly detection and classification tasks.
- Comprehensive evaluation of the proposed framework, demonstrating its potential to improve quality control in semiconductor manufacturing.
- Exploration of the broader implications of CLIP-based models for defect analysis in other industrial domains.

By bridging cutting-edge AI models and semiconductor defect analysis, this work aims to contribute to the development of intelligent quality control systems for next-generation semiconductor production.

2. Methods

2.1. Significance and Novelty

This study aims to detect seven types of anomalies in the semiconductor WM-811K dataset using the pre-trained Contrastive Language-Image Pre-training (CLIP) model's visual encoder. The novelty lies in applying a pre-trained visual model to a domain-specific problem, addressing challenges such as class imbalance, data preprocessing for high-dimensional image data, and efficient fine-tuning of pre-trained models.

2.2. Main Challenges and Solutions

- **Challenge: Class Imbalance and Bias Analysis**
  Solution: Data exploration, visualization, and stratified data splitting to mitigate bias.
- **Challenge: Efficient Feature Extraction**
  Solution: Utilize CLIP's pre-trained visual encoder to extract meaningful representations of images.

- **Challenge: Anomaly Categorization**
  Solution: Train a linear classifier on extracted features for accurate classification.

2.3. Main Figure

Figure ?? illustrates the workflow of the proposed anomaly detection model.

2.4. Reproducibility - Pseudocode

Algorithm 1 presents the steps for training and anomaly detection using the proposed method.

Algorithm 1 Anomaly Detection using CLIP's Visual Encoder

Image dataset D, labels L, pre-trained CLIP model (visual encoder), learning rate η, epochs E Split D into training (D_{train}) and testing (D_{test}) sets. Preprocess images using normalization and resizing to 224×224. Extract features using the CLIP visual encoder:

$$F = CLIP.encode\_image(D_{train})$$

Initialize a linear classifier C with weights W. Define loss function $\mathcal{L}$ as CrossEntropyLoss. epoch $e = 1$ to E $C \leftarrow$ Train classifier using F and L_{train}:

$$W \leftarrow W - \eta \nabla \mathcal{L}(C(F), L_{train})$$

Evaluate C on D_{test}. Output: Trained classifier C.

3. Experiments

3.1. Dataset

The WM-811K dataset, sourced from Kaggle, is a widely used dataset for anomaly detection in semiconductor manufacturing processes. It consists of $811,457$ wafer maps labeled with various defect patterns, which include seven primary categories:

- **Center**: Defects concentrated near the center of the wafer.
- **Donut**: Circular defect patterns around the wafer.
- **Edge-Loc**: Defects localized near the edge.
- **Edge-Ring**: Ring-shaped defects near the periphery of the wafer.
- **Loc**: Irregularly localized defects.
- **Random**: Randomly scattered defects across the wafer.
- **Scratch**: Linear defects caused by scratching.

Each wafer is represented as a binary image with dimensions 26×26, where the pixel value indicates whether a particular region is defective or not. For training, we resized all images to 224×224 to ensure compatibility with the pre-trained CLIP model. Data augmentation techniques such as random cropping and horizontal flipping were applied to improve model generalization.

3.2. Computing Resources

The experiments were conducted using Google Colab with the following specifications:

- **GPU**: NVIDIA Tesla T4 (Free Tier)
- **Framework**: PyTorch 2.0
- **Batch Size**: 16
- **Optimizer**: Adam optimizer with a learning rate of 1×10^{-4}

Despite the limited computational resources, the pre-trained CLIP model enabled efficient feature extraction and reduced training time for the anomaly classifier.

3.3. Experimental Design and Setup

The experimental design comprises the following stages:

3.3.1. 1. Data Preprocessing

- Verified and filtered image files with valid extensions
- Split the dataset into training and testing sets using an 80/20 ratio.
- Applied normalization and resized all images to 224×224 to align with CLIP's input requirements.
- Augmented the dataset with random cropping and flipping to improve robustness.

3.3.2. 2. Feature Extraction with CLIP

- Utilized the CLIP visual encoder to extract 512-dimensional features for each image.
- Enabled gradient computation for the classifier layer only, keeping CLIP's encoder frozen during training.

3.3.3. 3. Training and Evaluation

- A linear classifier was trained on the extracted features using CrossEntropyLoss.

- Evaluated the model's performance on the testing set after every epoch.
- Metrics such as accuracy, precision, recall, and F1-score were computed for each defect category to assess the model's efficacy.

3.4. Quantitative Results: Comparison with Baselines and State-of-the-Art

Table 1 compares the performance of the proposed method with existing state-of-the-art approaches for wafer defect classification.

Table 1. Performance comparison of wafer defect classification methods.

Method	Classification Accuracy (%)
YOLOv3-tiny	82.8
YOLOv3	94.4
YOLOv4	95.7
Proposed Method (CLIP)	98.0

The proposed method outperforms existing approaches, achieving the highest accuracy (98.0%) and demonstrating faster inference time due to the lightweight linear classifier.

3.5. Qualitative Results: Analysis of Success and Limitations

The proposed method's success is attributed to the following factors:

- **Effective Feature Extraction:** Leveraging CLIP's pre-trained visual encoder allows for robust feature extraction, leading to superior classification performance.
- **Simplified Training Process:** By using a lightweight linear classifier, the model avoids overfitting and remains computationally efficient.

However, the method has some limitations:

- **Lack of Multimodal Integration:** Unlike a true multimodal approach that integrates text and image data, this model uses only image-based features, limiting its ability to incorporate contextual metadata.
- **Reduced Dataset Size:** To balance the dataset and mitigate class imbalance, a significant portion of the data was removed, potentially restricting the model's ability to generalize to rare defect patterns.

Discussion

The proposed method successfully achieves high accuracy and efficient inference time due to the use of a pre-trained CLIP visual encoder and a lightweight classifier. However, the lack of multimodal data integration (e.g., text metadata) limits its potential for richer feature representation. Additionally, reducing the dataset size for balancing classes, while necessary to address bias, may have hindered the model's ability to handle rare defect types. Future work could explore multimodal approaches and larger, balanced datasets to further improve performance.

4. Future Directions

4.1. Expanding to Multimodal Approaches

While this study focuses on wafer defect classification using single-modal image data, an exciting future direction is to extend the framework to a multimodal approach. For instance:

- **Integration of Textual Data:** Semiconductor manufacturing processes often generate metadata, such as textual descriptions of defects, process parameters, or engineer annotations. Combining these textual data with image representations could enhance the model's ability to contextualize defect patterns and improve classification performance.
- **Leveraging CLIP's Full Capabilities:** The CLIP model's ability to encode both text and image modalities provides a natural avenue for creating a unified multimodal defect analysis system. For example, pairing wafer images with associated textual descriptions (e.g., "scratches near the edge") could improve the interpretability and robustness of the model.

4.2. Domain Adaptation for Broader Applications

To extend the applicability of the proposed framework, domain adaptation techniques could be explored. For instance:

- **Other Industrial Domains:** The methodology could be applied to other fields requiring anomaly detection, such as medical imaging, automotive manufacturing, or aerospace quality control.
- **Cross-Dataset Training:** Training the model across multiple wafer datasets or defect datasets from different manufacturers may improve its generalizability and real-world performance.

4.3. Improving Model Efficiency

Given the computational constraints of real-world manufacturing environments, future work could focus on optimizing the model for deployment:

- Reducing computational complexity through model compression or quantization.

- Exploring real-time inference techniques for faster defect detection.

4.4. Unsupervised and Semi-Supervised Learning

For rarer defect categories with limited labeled samples, semi-supervised or unsupervised techniques could play a crucial role in improving anomaly detection accuracy:

- Implementing clustering methods to identify new or evolving defect patterns.
- Exploring contrastive learning frameworks for representation learning without extensive labeled data.

References